



SensibleAI Studio Guide

Copyright © 2026 OneStream Software LLC. All rights reserved.

All trademarks, logos, and brand names used on this website are the property of their respective owners. This document and its contents are the exclusive property of OneStream Software LLC and are protected under international intellectual property laws. Any reproduction, modification, distribution or public display of this documentation, in whole or part, without written prior consent from OneStream Software LLC is strictly prohibited.

Table of Contents

Overview 1

 Explore 1

 Run 2

Setup and Installation 3

 Dependencies 4

 Installation 4

 Set Up SensibleAI Studio 4

Settings 7

 Global Settings 7

 Uninstall 8

 Solution Info 9

Navigate in SensibleAI Studio 11

 SensibleAI Studio Home Page 11

 SensibleAI Studio Sections 12

Table of Contents

Toolbar Icons	13
Routines	15
Routine Instance	15
Routine Method	15
Routine Constructor	16
Memory Capacity	16
Stateful Routine	17
Stateless Routine	24
Routine Instances	27
Routine Instance Actions	27
Create	27
Invoke	27
Edit	31
Delete	32
Routine Runs	33
Routine Run Actions	33

Table of Contents

Rerun	33
Copy	34
Edit	35
Delete	36
Cancel	37
Routine Instance and Routine Run Views	38
Summary	38
Artifacts	41
Documentation	43
Package Specifications and Usage Limits	44
SensibleAI Studio "Essentials" Package	44
Description	44
Offering Includes	44
Usage Limits	45
SensibleAI Studio "Standard" Package	46
Description	46

Table of Contents

Offering Includes46

Usage Limits47

Definitions48

Overview

This document details the SensibleAI Studio user interface, including functionality and requirements of each page. Information includes:

- How to interact with each section and page in the solution.
- How to navigate all routines in the Routine Library and create a Routine Instance.
- How to invoke and view a Routine Method Run.

Explore

The first major section of SensibleAI Studio is the Explore page. On this page, users can select from a list of all the available Routines and choose which to use for creating a new Routine Instance.

Click on Routine Name to see more information on the right side of the page; pages include Definition, Signature, and Metadata.

- **Definition:** This section displays details about the Routine, its use cases, and the creation date.
- **Signature:** This section displays details about methods, required configuration inputs, outputs, and default memory capacity.
- **Developer Docs:** This section provides creators and power users with the necessary information to invoke the selected routine programmatically. This information includes raw method names and their corresponding artifact qualified keys, if applicable.

Routine Library: This is where the individual Routines are stored within the solution. The total number of Routines here increases as future iterations of SensibleAI Studio are developed and released.

Run

The **Run** page is the next major section of SensibleAI Studio. Here, users can create new Routine Instances and Routine Runs or interact with existing ones. Users can also click the self-icon in each section to display only their personal instances or runs.

Selecting a Routine Instance will automatically filter the Routine Runs Activity grid to show only the runs associated with that instance. Additional filters for Routine Runs such as **Stateful**, **Stateless**, **Today**, **Completed**, **Running**, and **Queued** can be applied individually or in combination to further customize the view.

By selecting a Routine Instance or Routine Run, users can view its **status**, **summary**, **artifacts**, and **documentation** on the right side of the page. Depending on the selection, users also have access to a range of available actions tailored to either the instance or the run.

For more information, see the [Routine Instances](#) and [Routine Runs](#) sections.

Setup and Installation

This section contains details for planning, configuring, and installing the SensibleAI Studio solution. Before you install the solution, familiarize yourself with these details.

Solution Info

Solution Version:

[STU-SV140-XPfv4.3.0-PV930](#)

Xperiflow Version:

[4.3.0](#)

Minimum Routine Memory Capacity:

[1 GB](#)

Maximum Concurrent Routine Runs:

[10](#)

Total Environment Memory Capacity:

[10 GB](#)

See OneStream Solution Modification Considerations.

Dependencies

Component	Description
OneStream 9.3.0 or later	Minimum OneStream Platform version required to install this version of SensibleAI Studio.
Xperiflow 4.3.0 or later	Minimum version required to install this version of SensibleAI Studio.
Xperiflow Business Rules V230 (XBR)	External API client library to allow STU to interface with the Xperiflow Engine. The required version of XBR is packaged with all STU versions.

Installation

The *Getting Started with Solutions Guide* provides general instructions to install and configure OneStream Solutions. It also includes the recommended display settings, package file name information, advice for solution database migration, and considerations for modifications to solutions.

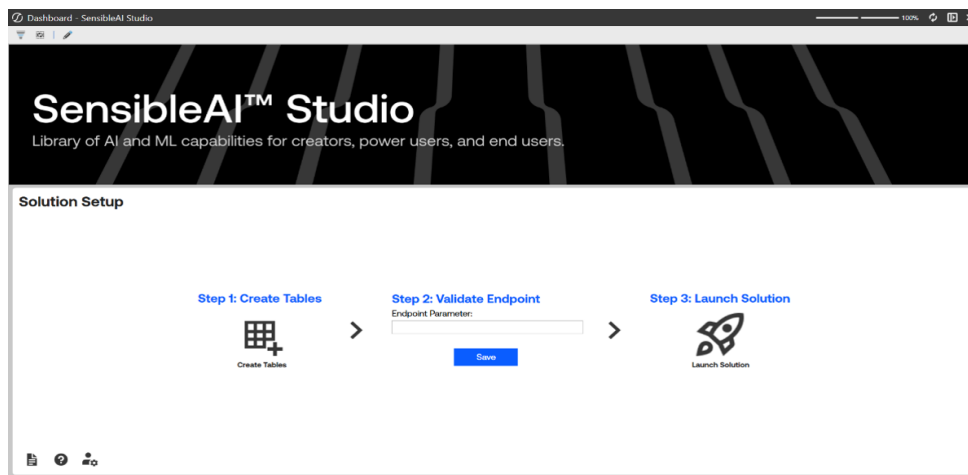
See the *Getting Started with Solutions Guide*.

Set Up SensibleAI Studio

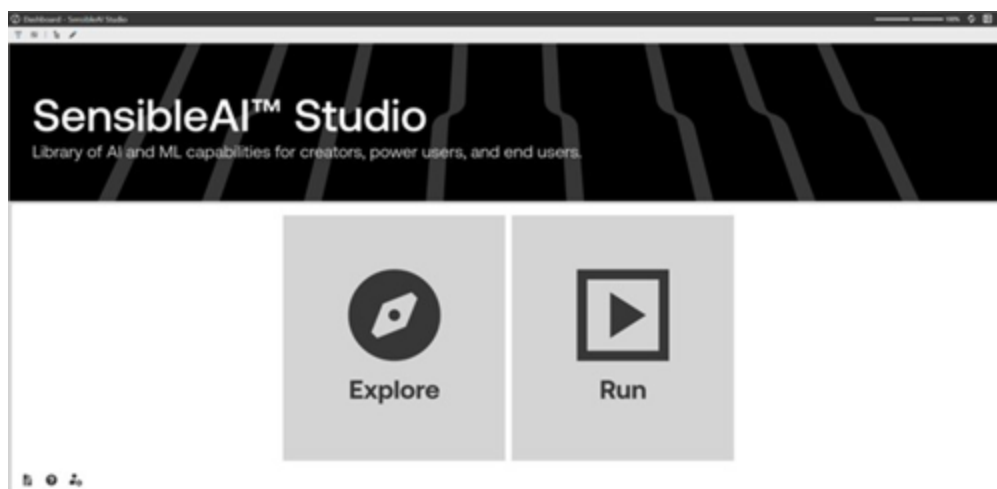
Follow these steps to set up SensibleAI Studio:

Setup and Installation

1. Download the SensibleAI Studio Solution from the OneStream Solution Exchange.
2. After the OneStream support team ensures that the proper contract is in place, a link is sent to download the SensibleAI Studio solution. To complete the setup, the user must now set the proper endpoint parameter and launch the solution.
3. Follow the outlined Solution Setup steps:



- a. Create Tables
 - b. Validate Endpoint (will vary based on the environment)
 - c. Launch Solution
4. When a user reaches the Home page after clicking Launch, SensibleAI Studio has been successfully set up correctly:



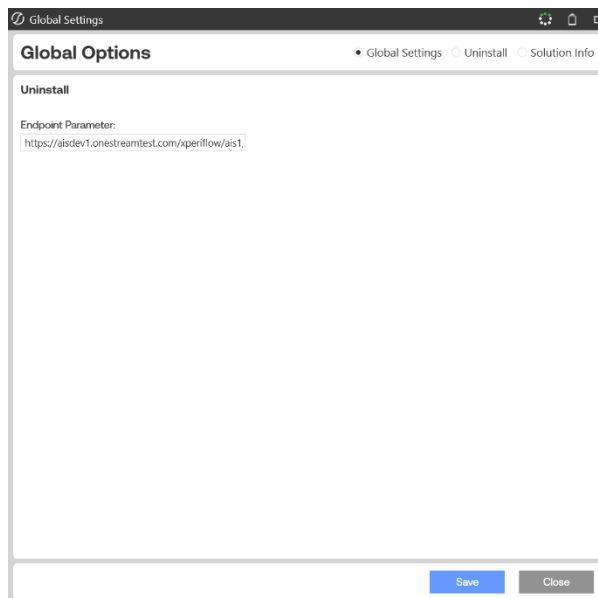
Settings

To access the Global Options page, click **Settings**() in the bottom left corner of the **Explore** page or the **Run** page.

Global options include:

- [Global Settings](#)
- [Uninstall](#)
- [Solution Info](#)

Global Settings



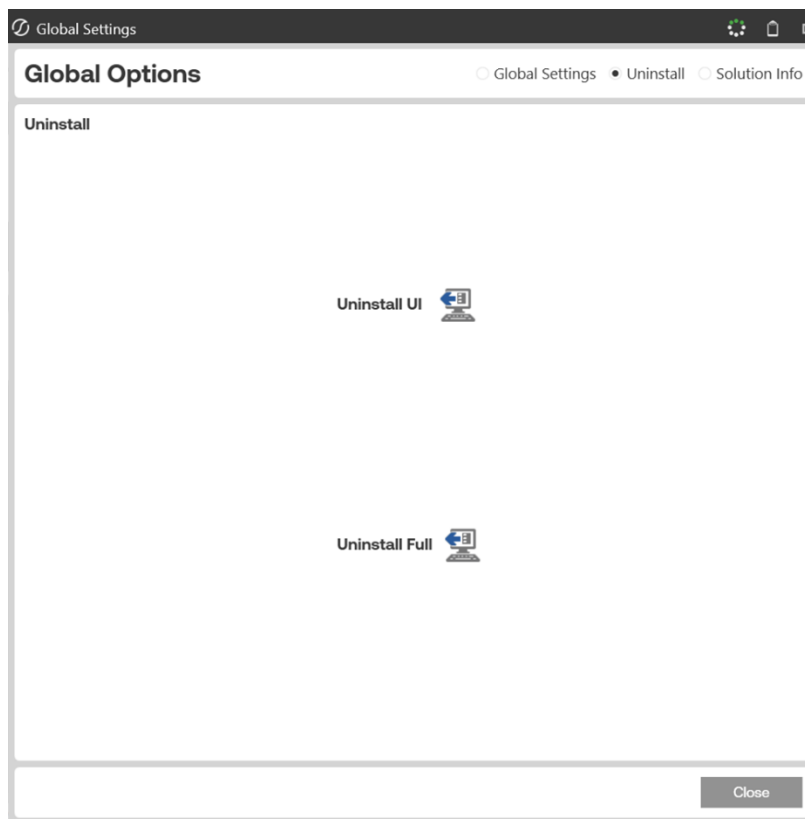
Settings

Endpoint Parameter

Predefined endpoint to access the application.

CAUTION: Do not make changes to this value unless instructed to do so.

Uninstall



There are two uninstall options:

CAUTION: Uninstall procedures are irreversible.

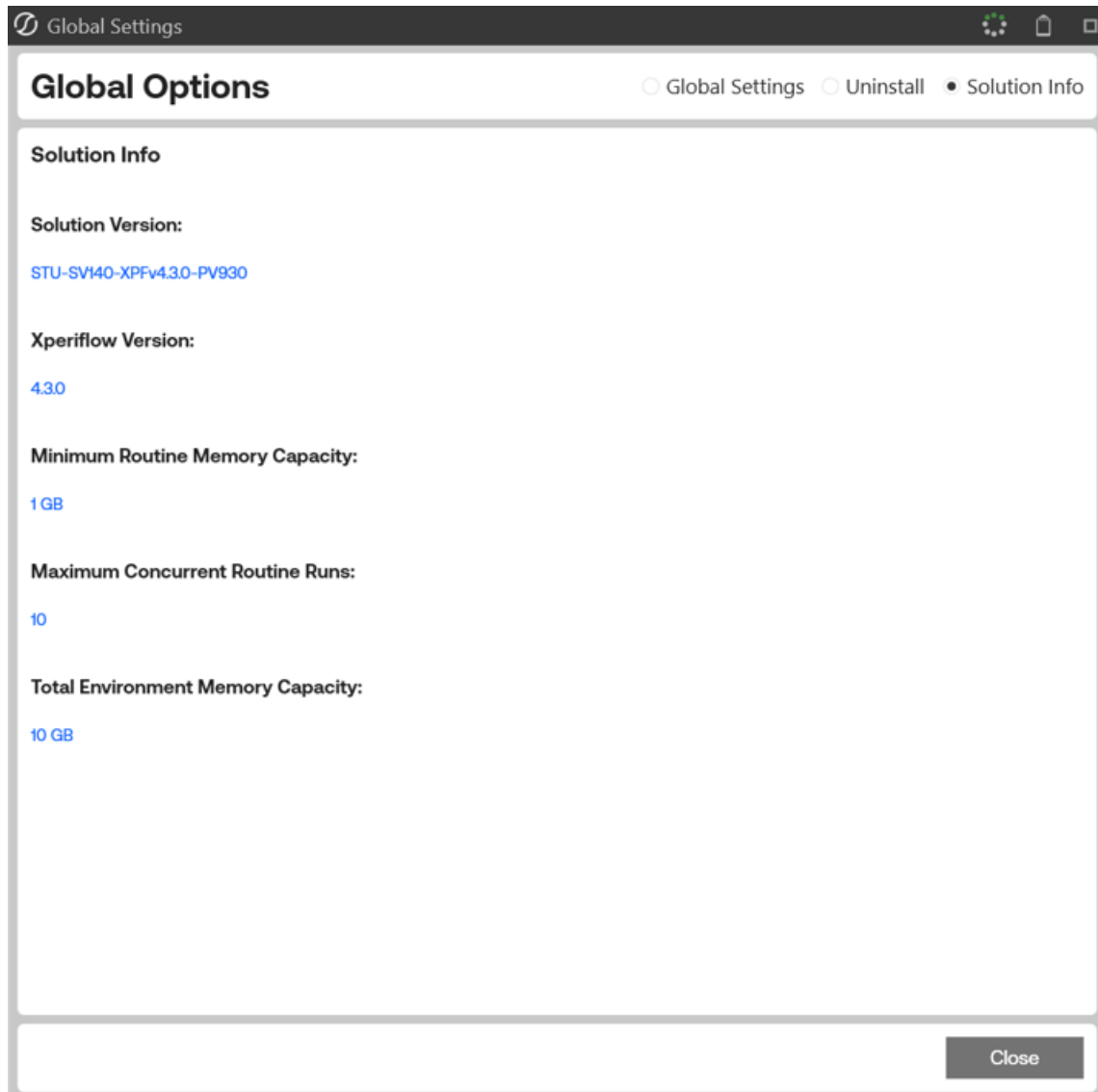
Uninstall UI removes SensibleAI Studio, including related dashboards and business rules, but leaves the database and related tables in place. Choose this option if you want to accept a SensibleAI Studio update without removing data tables.

Uninstall Full removes all related data tables, data, SensibleAI Studio dashboards, and business rules. Choose this option to completely remove SensibleAI Studio or to perform an upgrade that is so significant in its changes to the data tables that this method is required.

Solution Info

The Solution Version is comprised of the Solution Code, Solution Version, Xperiflow Version, and OneStream platform version (Solution Code-Solution Version-Xperiflow Version-OneStream Platform Version).

Settings

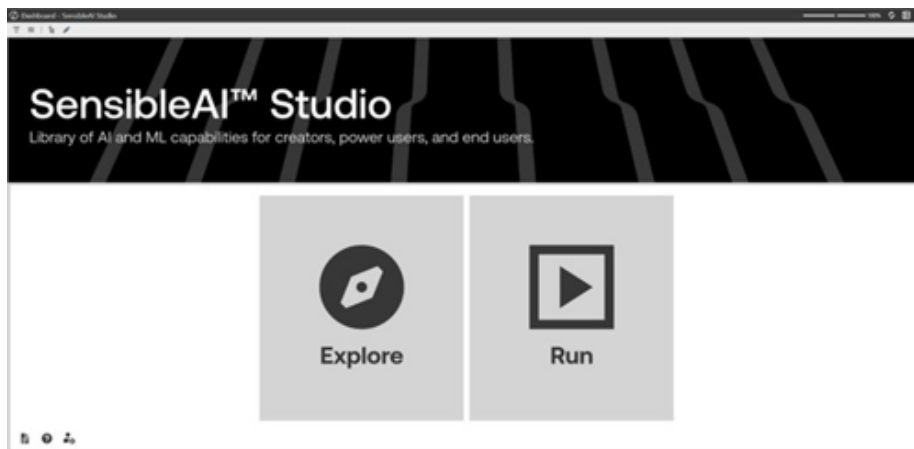


Navigate in SensibleAI Studio

The following sections describe the ways to navigate the SensibleAI Studio solution.

SensibleAI Studio Home Page

The Home page displays a clickable interface prompting the user to launch either the Explore page or the Run page.



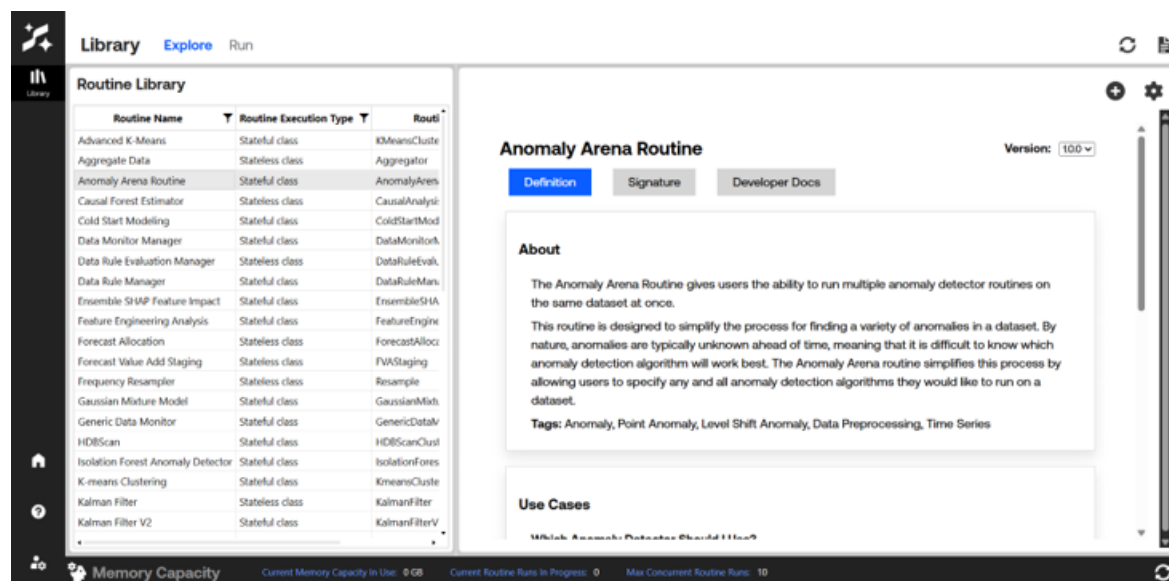
Once launched, the **Refresh** on the top right of the screen allows users to refresh the current page. The **Home** button to the right brings users back to the Home page.

SensibleAI Studio Sections

The left navigation panel includes a single section, **Library**. The top left corner of the upper toolbar shows the pages available in the selected section. Below are the available pages and their components:

Explore Page

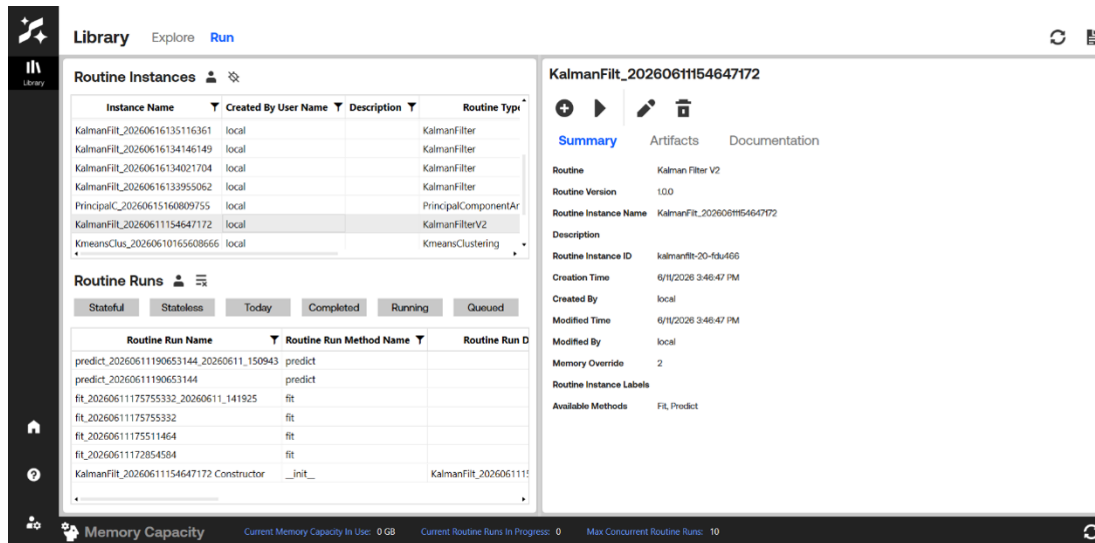
Within the Routine Library, click a Routine Name to view documentation for all versions of that Routine. Documentation includes a Definition section, Signature information, and Developer Docs. Users can create a Routine Instance or access the Routine Settings configurations.



Run Page

Interact with existing Routine Instances and Routine Runs, or create new ones as needed. Whether or not an instance or run is selected, users can access additional details in the **Summary** section, view method-run outputs in the **Artifacts** section, and explore routine-specific information in the **Documentation** section. For more information, see [Routine Instances](#) and [Routine Runs](#) sections.

Navigate in SensibleAI Studio



Toolbar Icons

Icon	Description
 Home	Navigates to the Home page.
 Help	Opens the SensibleAI Studio User Guide.
 Settings	Opens the SensibleAI Studio settings. Configure Global Settings options and uninstall the solution.

Icon	Description
 Refresh	Refreshes the current page.
 Activity Log	Displays all job activity from the AI Services engine for jobs executed within SensibleAI Studio.

Each section page includes **Refresh** and **Activity Log** buttons at the top right and **Home**, **Help**, and **Settings** buttons at the lower left.

Routines

A Routine is an AI or ML capability catered toward solving a particular use case. There is a plethora of Routines housed within SensibleAI Studio and more to come in future iterations and updates to the solution. All available Routines are found in the Explore page within the Routine Library.

Example: Some examples of Routines include Forecast Allocation, Source Data Analysis, ML Regression, K-Means Clustering, and so on.

Routine Instance

A user-created routine object that is required to execute a Routine method. Routine Instances are created from both the Explore and Run pages.

Routine Method

A discrete, executable function that is specific to a given Routine. Routine methods are run by using the Invoke or Rerun buttons on the Run page.

Routine Constructor

A function specific to a given [Stateful Routine](#) that encapsulates logic to be executed before any additional Routine Methods can be invoked. Constructor Runs are automatically triggered upon the creation of a Routine Instance and may only be executed once per instance. Although Constructor Runs appear in the Routine Runs section of the Run page alongside other Routine Method Runs, certain actions such as Rerun and Delete are prohibited for Constructor Runs.

Memory Capacity

- The amount of memory, in gigabytes, that a Routine Run may use. Users may configure memory capacity as needed, but Routine Methods default to 2GB.
- Memory Capacity Rules
 - SensibleAI Studio users can configure the memory capacity levels. The following list outlines the order of precedence for determining the memory capacity of a Routine Run:
 - The user-defined memory capacity set during the creation of the Routine Run, if it was set.
 - The user-defined memory capacity specifically set on the Routine Method, if it was set.
 - The routine developer-defined memory capacity for that specific Routine Method, if it was set.
 - The user-defined memory capacity set during creation of the Routine Instance, if it was set.

- The user-defined defaults memory capacity for the specific Routine, if it was set.
- The routine developer-defined memory capacity for that specific Routine, if it was set.
- Default to the global default routine memory if none of the above were set.

Stateful Routine

A Stateful Routine is a routine that maintains internal state over time. These routines keep track of internal data that may be updated when its Routine Methods are invoked. Stateful Routines will typically require input parameters when creating a Routine Instance.

Example: Anomaly Arena Routine maintains a list of models fit to the provided input data. Below shows how to navigate the component workflow for a Stateful Routine.

The screenshot displays the SensibleAI Studio interface. On the left, the 'Routine Library' is shown with a table of routines. The 'Anomaly Arena Routine' is highlighted. On the right, the details for the 'Anomaly Arena Routine' are displayed, including its version (1.0.0) and tabs for 'Definition', 'Signature', and 'Developer Docs'. The 'About' section describes the routine's purpose: to simplify the process of finding anomalies in a dataset by allowing users to specify any and all anomaly detection algorithms. The 'Use Cases' section lists various applications: Anomaly, Point Anomaly, Level Shift Anomaly, Data Preprocessing, and Time Series.

Routine Name	Routine Execution Type	Routine
Advanced K-Means	Stateful class	KMeansCluste
Aggregate Data	Stateless class	Aggregator
Anomaly Arena Routine	Stateful class	AnomalyAren
Causal Forest Estimator	Stateless class	CausalAnalyti
Cold Start Modeling	Stateful class	ColdStartMod
Data Monitor Manager	Stateful class	DataMonitorM
Data Rule Evaluation Manager	Stateless class	DataRuleEval
Data Rule Manager	Stateful class	DataRuleMan
Ensemble SHAP Feature Impact	Stateful class	EnsembleSHAP
Feature Engineering Analysis	Stateful class	FeatureEngin
Forecast Allocation	Stateless class	ForecastAlloc
Forecast Value Add Staging	Stateless class	FVAStaging
Frequency Resampler	Stateless class	Resample
Gaussian Mixture Model	Stateful class	GaussianMix
Generic Data Monitor	Stateful class	GenericDataM
HDBScan	Stateful class	HDBScanClus
Isolation Forest Anomaly Detector	Stateful class	IsolationFore
K-means Clustering	Stateful class	KmeansCluste
Kalman Filter	Stateless class	KalmanFilter
Kalman Filter V2	Stateful class	KalmanFilterV

Anomaly Arena Routine Version: 1.0.0

About

The Anomaly Arena Routine gives users the ability to run multiple anomaly detector routines on the same dataset at once.

This routine is designed to simplify the process for finding a variety of anomalies in a dataset. By nature, anomalies are typically unknown ahead of time, meaning that it is difficult to know which anomaly detection algorithm will work best. The Anomaly Arena routine simplifies this process by allowing users to specify any and all anomaly detection algorithms they would like to run on a dataset.

Tags: Anomaly, Point Anomaly, Level Shift Anomaly, Data Preprocessing, Time Series

Use Cases

Which Anomaly Detectors Should I Use?

Routines

Select your Routine from the list of available Routines. For this example, **The Anomaly Arena Routine** is used. Click **Create** at the top right of the screen.

The screenshot shows a window titled "Routine Instance Creation and Constructor Run". At the top, there is a breadcrumb navigation: "1. Create > 2. Parameters > 3. Run". The main content area is titled "Anomaly Arena Routine". It contains several form fields: a text input for "(Optional) Name" with the value "Anomaly Arena Routine Instance Demo" and a subtext "Provide a name for this Routine Instance."; a text area for "(Optional) Description" with a subtext "Provide a description for this Routine Instance."; a dropdown menu for "Memory Capacity" set to "2 GB" with a subtext "Define the memory capacity for this Routine Instance."; a dropdown menu for "Parameters Input Method" set to "Interactive" with a subtext "Select the way in which to fill out the parameters."; a dropdown menu for "Routine Version" set to "1.0.0" with a subtext "Select a specific routine version to run."; and a dropdown menu for "(Optional) Routine Instance Labels" with a subtext "Select an Existing Routine Label". At the bottom right, there are two buttons: "Next" (highlighted in blue) and "Cancel".

Name your Routine Instance, update memory capacity if necessary, and choose the desired Routine Version. The latest version will always be selected by default.

NOTE: The default size is 2GB. Increase memory capacity size for larger datasets.
Scroll down.

Routines

Routine Instance Creation and Constructor Run

1. Create > 2. Parameters > 3. Run

Anomaly Arena Routine

Provide a description for this Routine Instance.

Memory Capacity
2 GB

Define the memory capacity for this Routine Instance.

Parameters Input Method
Interactive

Select the way in which to fill out the parameters.

Routine Version
1.0.0

Select a specific routine version to run.

(Optional) Routine Instance Labels

Select an Existing Routine Label

Select from the list of existing labels.

Add a new Routine Label

Time Series Anomaly Detection

Click the Create button to add a new label.

Routine Labels: Time Series Anomaly Detection

Next Cancel

Select one or more existing labels for your Routine Instance. To create a new label, enter the desired name in the text box and click the Create button to the right of it. All newly created or selected labels will show next to the Routine Labels text on the last line. To remove a newly created label, ensure the name is correctly spelled in the text box and click the Delete button to the right of the text box. To remove an existing label that has been selected, simply uncheck it from the dropdown menu. To delete all labels, click the Delete button on the last line. Click **Next**.

Routines

The screenshot shows a software window titled "Routine Instance Creation and Constructor Run". At the top, a breadcrumb trail indicates the current step: "1. Create > 2. Parameters > 3. Run". The main heading is "Anomaly Arena Routine". Below this, the subtitle is "Create: Anomaly Arena Routine Instance Demo Constructor".

The interface is divided into two main sections. The left section, titled "Connection Key", contains a dropdown menu with "sql-server-shared" selected. Below the dropdown is a small blue information icon and the text "The MetaFileSystem connection key." The right section contains explanatory text: "Select a MetaFileSystem connection key to that corresponds to a file system.", "'Routine' is where metadata, routine specific shared data, and data generated by routines is stored.", and "'Shared' is where any freeform sharable data is stored."

Below these sections is a horizontal separator with the text "Source Data Definition - Source Connection - Connection - Choose File Path". Under this, the "File Path" section has a dropdown menu with "SampleSalesData.csv" selected, followed by a blue information icon and the text "The full file path to the file to ingest." The "MetaFileSystem Files" section on the right displays a list of files: "SampleSalesData.csv", "mrbig.parquet", and "semicolon.csv".

At the bottom of the window, there are navigation controls: two circular arrows (back and forward) and three buttons labeled "Previous", "Next", and "Cancel".

Navigate through the Component Workflow while making selections based on the specific needs of your use case. Many steps may be left to default configurations, but for steps such as **Table Selection**, users need to select their specific table from the drop-down menu.

Routines

1. Create > 2. Parameters > 3. Run

Anomaly Arena Routine

Create: Anomaly Arena Routine Instance Demo Constructor

Dimension Columns
The columns to use as dimensions.

- part_number x
- part_family x
- store_id x

+ Add More Items

Date Column
The column to use as the date.

- date

Value Column

- amount

Data Preview (Top 100 Rows)

Id	Part_number	Business_unit	Part_family
2031344977_GSPR06_MI	2031344977	Clubs	Acme
2031344977_GSPR06_MI	2031344977	Clubs	Acme
2031344977_GSPR06_MI	2031344977	Clubs	Acme
2031344977_GSPR06_MI	2031344977	Clubs	Acme
2031344977_GSPR06_MI	2031344977	Clubs	Acme
2031344977_GSPR06_MI	2031344977	Clubs	Acme
2031344977_GSPR06_MI	2031344977	Clubs	Acme

Page Size: 100 Page 1 of 104

Previous Next Cancel

Continue to complete each step of the Component Workflow, selecting **Dimension**, **Date**, and **Value** columns that align with your selected dataset. If adding in a **Feature** dataset, ensure to select **Add** from the drop-down menu. If you are not adding features, select **Continue**. Ensure to include a **Group Name** when prompted.

Routines

The screenshot shows a software window titled "Routine Instance Creation and Constructor Run". At the top, a breadcrumb trail reads "1. Create > 2. Parameters > 3. Run". The main content area is titled "Anomaly Arena Routine" and contains a sub-header "Create: Anomaly Arena Routine Instance Demo Constructor". Below this, there is a section for "Anomaly Detector Configurations" with a text prompt "Choose the anomaly detector method to use." and a drop-down menu currently showing "Continue". A small informational icon is present below the menu. The "Review" section at the bottom states, "You have completed the workflow. Please review and submit your inputs." At the very bottom of the window are three buttons: "Previous", "Next", and "Cancel".

Update selections to configure one or more anomaly detectors. Once finished with adding anomaly detectors, select **Continue** from the drop-down menu to move on.

Routines

1. Create > 2. Parameters > 3. Run

Anomaly Arena Routine

Routine Instance Information

Routine	AnomalyArena
Routine Version	1.0.0
Routine Instance Name	Anomaly Arena Routine Instance Demo
Constructor Run Name	Anomaly Arena Routine Instance Demo Constructor
Description	
Routine Instance ID	anomaly-arena-og26gc
Constructor Run ID	anomaly-arena-0763cj
Memory Capacity	2
Routine Instance Labels	Time Series Anomaly Detection

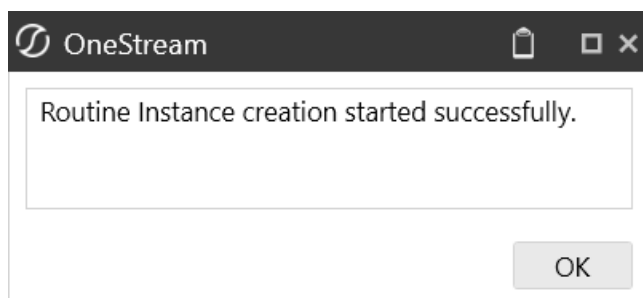
Provided Input Parameters

```
{
  "source_data_definition": {
    "data_connection": {
      "tabular_connection": {
        "connection_key": "sql-server-shared",
        "file_path": "SampleSalesData.csv"
      }
    }
  },
  "dimension_columns": [
    "CustomerSegment",
    "Region"
  ]
}
```

The parameter inputs needed to run this constructor. This is resolved back to user in case user would like to create to run noninteractively in the future.

Previous Submit Cancel

After completing the Component Workflow, click **Next** to review the details of your Routine Instance. To make adjustments, click **Previous** to navigate back to the desired step. To create your Routine Instance, click **Submit** on the final Review step. A notification will pop up notifying you that Routine Instance creation has begun. You can view your Routine Instance under the Routine Instances panel on the Run page. The associated Constructor Run will also be visible under Routine Runs.

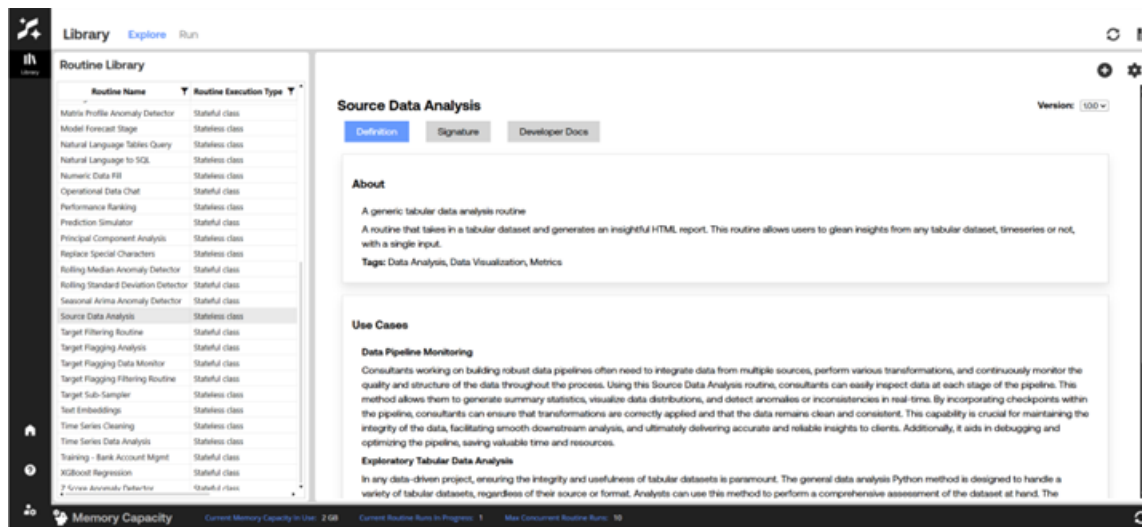


NOTE: The Component Workflow for Stateful Routines will vary based on the specific routine selected; each has its own configurations specific to the type of Routine. Inputs may differ slightly from what has been shown in this walkthrough.

Stateless Routine

A routine that does not maintain any sort of internal state. Stateless Routine's methods can be described as static. Unlike Stateful Routines, Stateless Routines do not require a constructor do not require any input parameters when creating a Routine Instance.

Example: Source Data Analysis – does not require input parameters; no internal state.



Navigate to the **Explore** page of SensibleAI Studio to begin this process. Select a routine. For this example, the **Source Data Analysis Routine** will be used. Click **Create** at the top right of the screen.

Routines

1. Create

Source Data Analysis

(Optional) Name
SDA Routine Instance
Provide a name for this Routine Instance.

(Optional) Description

Provide a description for this Routine Instance.

Memory Capacity
2 GB
Define the memory capacity for this Routine Instance.

Routine Version
1.0.0
Select a specific routine version to run.

(Optional) Routine Instance Labels

Select an Existing Routine Label
Select from the list of existing labels.

Add a new Routine Label
+ -

Finish Cancel

Name your Routine Instance and include a description (optional). Ensure that the correct Routine Version is selected.

NOTE: Due to the nature of this being a Stateless Routine, there is no constructor method that will be run upon Routine Instance creation. Therefore, there will **not** be a Component Workflow.

Routines

1. Create

Provide a name for this Routine Instance.

(Optional) Description

Provide a description for this Routine Instance.

Memory Capacity

2 GB

Define the memory capacity for this Routine Instance.

Routine Version

1.0.0

Select a specific routine version to run.

(Optional) Routine Instance Labels

Select an Existing Routine Label

Select from the list of existing labels.

Add a new Routine Label

Data Analysis

Click the Create button to add a new label.

Routine Labels: Data Analysis

Finish Cancel

Select one or more existing labels for your Routine Instance. To create a new label, enter the desired name in the text box and click the Create button to the right of it. All newly created or selected labels will show next to the Routine Labels text on the last line. To remove a newly created label, ensure the name is correctly spelled in the text box and click the Delete button to the right of the text box. To remove an existing label that has been selected, simply uncheck it from the dropdown menu. To delete all labels, click the Delete button on the last line.

After clicking the **Submit** button, click on the **Run** page at the top of the screen to locate and interact with this Routine Instance.

Routine Instances

All Routine Instances are listed in the Routine Instances table on the Run page. To view only the instances you have created, click the **user-icon** ().

When you select a Routine Instance, the Routine Runs table will automatically update to show only the runs associated with that instance. To clear your selection, click the **deselect icon** (), this will also remove the filter from the Routine Runs table and display all runs.

Routine Instance Actions

When a Routine Instance has been selected on the Run page, the following actions may be completed: Create, Invoke, Edit, or Delete.

Create

The Create button allows users to generate a new Routine Instance based on the selected Routine Type. It works the same as the **Create** button on the **Explore** page. For detailed instructions on creating Routine Instances, see [Stateless Routine](#) and [Stateful Routine](#) sections.

Invoke

The **Invoke** button starts a new Routine Method Run for the selected Routine Instance.

Example: For this example, a Routine Instance for **Source Data Analysis** is used.

Routine Instances

1. Once the Routine Instance is selected, click the Invoke button on the right.

SDA Routine Instance



Invoke a new Method Run for the selected Routine Instance.

Summary Artifacts Documentation

Routine Source Data Analysis

Routine Version 1.0.0

Routine Instance Name SDA Routine Instance

2. Select the **Source Data Analysis** method then click **Next**.
3. Name your Routine Run and ensure that the **Input Method** is set to **Interactive**. Artifact statistics and previews will automatically be generated by default; set either option to No if you do not want statistics or previews to be generated.

Parameters Input Method

Interactive ▼

Select the way in which to fill out the parameters.

Memory Capacity

2 GB ▼

Define the memory capacity for this method run.

Include Artifact Statistics

Yes ▼

Choose if you would like to include artifact statistics for this method run.

Include Artifact Previews

Yes ▼

Choose if you would like to include artifact previews for this method run.

NOTE: Setting the Input Method to Interactive will task users with completing a Component Workflow process.

Routine Instances

4. Select or create new labels for the Routine Method Run and click **Next**.
5. Complete the interactive workflow process. These steps follow the same logic as with other Component Workflow tasks; users need to select a **Database Connection**, a **Database Resource** or **File Connection Key**, and a **Database Name** or **File Path** to start.

The screenshot shows a web-based interface for creating an SDA Routine Instance. The window title is 'Method Run Invocation'. The breadcrumb navigation shows the current step is '3. Parameters' within a sequence of '1. Method', '2. Create', and '4. Run'. The main heading is 'SDA Routine Instance (Method: Source Data Analysis)'. Below this, it says 'Create: sourcedata_20260624172926799'. A sub-section titled 'Data Connection - Connection - File Connection Key' contains a 'Connection Key' dropdown menu with 'sql-server-shared' selected. To the right of the dropdown, there is explanatory text about 'Partitioned MetaFileSystem Connection' and 'OneStream Cube View'. At the bottom of the window, there are three buttons: 'Previous' (disabled), 'Next' (active), and 'Cancel' (disabled).

Method Run Invocation

1. Method > 2. Create > 3. Parameters > 4. Run

SDA Routine Instance (Method: Source Data Analysis)

Create: sourcedata_20260624172926799

The connection type to use to access the source data.

Partitioned MetaFileSystem Connection: A connection to a set of MetaFileSystem tabular files. Note that it is assumed that all files in the directory are of the same tabular type and have the same schema. The supported file types are .csv, .tsv, .psv, and .parquet. The connection will only look at the first inner level of the directory for the files. Any subdirectories are ignored.

OneStream Cube View: A connection to a pre-configured OneStream cube view. The cube view is extracted fresh from the cube at each routine execution.

Data Connection - Connection - File Connection Key

Connection Key

sql-server-shared

The MetaFileSystem connection key.

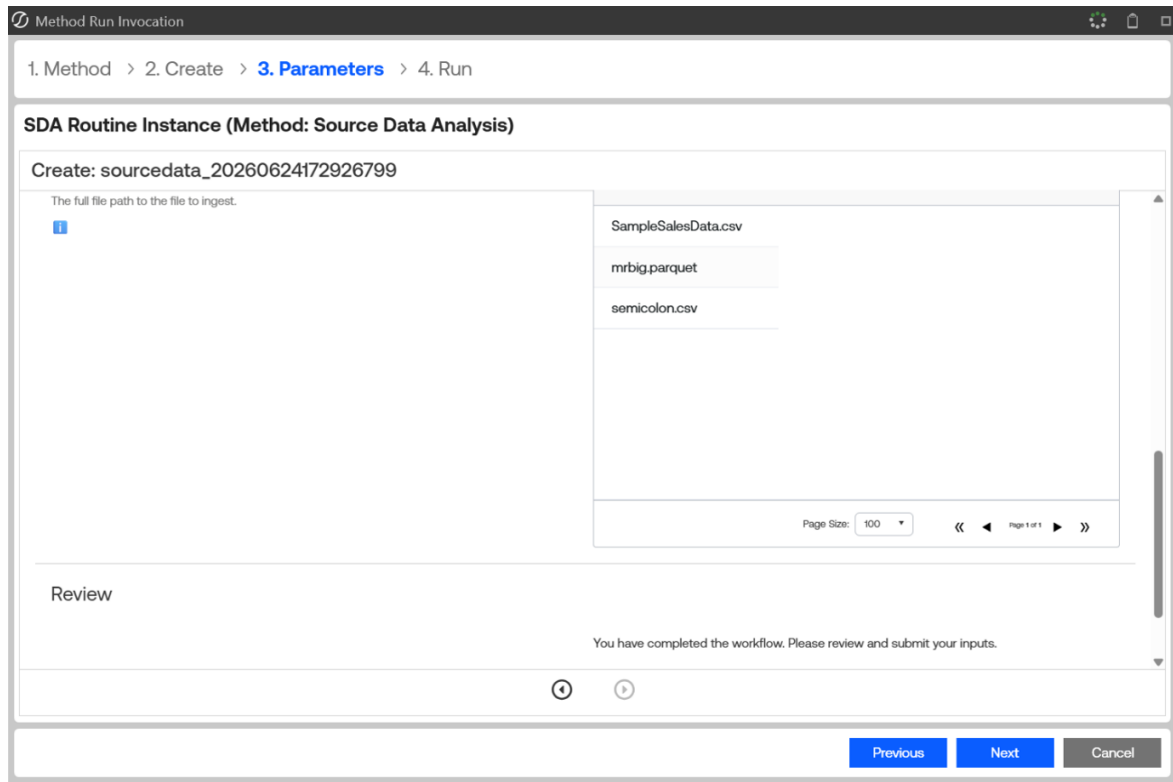
Select a MetaFileSystem connection key to that corresponds to a file system.

'Routine' is where metadata, routine specific shared data, and data generated by routines is stored.

'Shared' is where any freeform sharable data is stored.

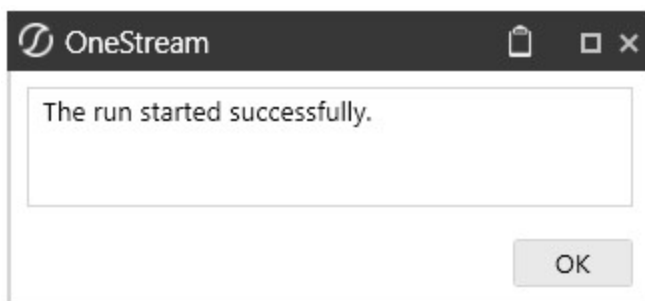
Previous Next Cancel

Routine Instances



Once you hit the Review step, you have completed the Component Workflow. Click **Next**.

6. Review input parameters and details of your Source Data Analysis Routine Method Run and click **Submit** to start the run.



After submitting, it may take a few minutes for the run to complete.

Routine Instances

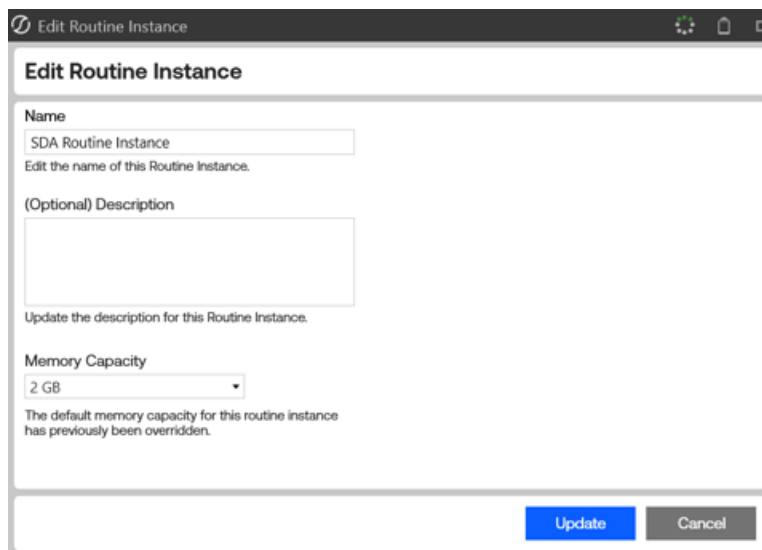
NOTE: The Component Workflow will vary based on the routine method selected. Inputs may differ slightly from what has been shown in this walkthrough.

Additionally, Json inputs may be used to run a new method in place of the Component Workflow by setting the **Input Method** to Json when creating a new method run. After completing the Component Workflow for a given method, the Json input can be copied on the Review page before executing the run. After the run has started, this Json input can also be accessed by clicking the [Copy](#) button.

Edit

Users update the Routine Instance Name and Routine Instance Description by clicking the **Edit** button.

NOTE: It is intended that the Routine Run Name and Routine Run Description do not change.

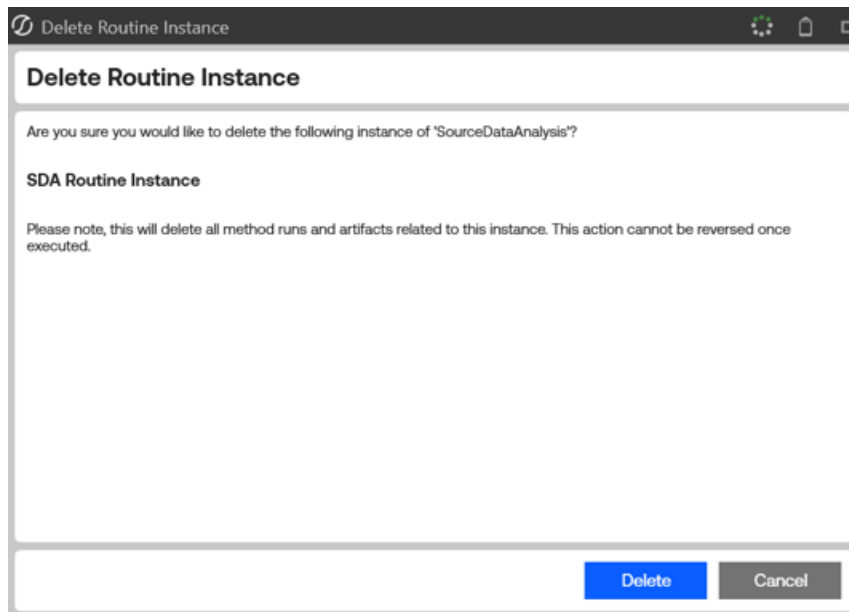


The screenshot shows a dialog box titled "Edit Routine Instance". It contains three main sections: "Name" with a text input field containing "SDA Routine Instance" and a hint "Edit the name of this Routine Instance."; "(Optional) Description" with a larger text area and a hint "Update the description for this Routine Instance."; and "Memory Capacity" with a dropdown menu set to "2 GB" and a hint "The default memory capacity for this routine instance has previously been overridden." At the bottom right are "Update" and "Cancel" buttons.

Delete

Users delete selected routine instances entirely by clicking the **Delete** button.

CAUTION: All Routine Method Runs and Artifacts attached to this Routine Instance will also be deleted. All deletions cannot be reversed.



Routine Runs

A Routine Run is the term used to describe any routine-specific job that is executed by a user.

Each row in the Routine Runs table, located on the Run page of SensibleAI Studio, represents a Routine Run. This includes **Constructor Runs**, which are triggered immediately after the successful creation of a Routine Instance, as well as **Method Runs**, which are specific to a particular Routine Instance and vary depending on the Routine. Available actions depend on whether a Routine Constructor Run, or a Routine Method Run has been selected.

To view only the runs you have created, click the **user-icon** (). Additional filters for Routine Runs such as Stateful, Stateless, Today, Completed, Running, and Queued can be applied individually or in combination to further customize the view. To clear all Routine Run filters, click the **clear filters icon** ().

Routine Run Actions

When a Routine Method Run has been selected, the following actions may be completed: Rerun, Copy, Edit, Delete, or Cancel.

Rerun

Users choose to rerun their selected Routine Method Run by clicking the **Rerun** button. The system will automatically update the Routine Run Name by appending the current date and time to the original name. Users can modify input parameters by editing the provided JSON snippet, and optionally adjust the memory capacity or description for this new run.

Routine Runs

Rerun

Name
sourcedata_20260624172926799_20260624_1409
Set a new name for this method rerun.

(Optional) Description
Provide an updated description for this method rerun.

Memory Capacity
2 GB
Set the memory capacity for this method rerun.

Input Parameters

```
{
  "data_connection": {
    "tabular_connection": {
      "connection_key": "sql-server-shared",
      "file_path": "SampleSalesData.csv"
    }
  }
}
```

Run **Cancel**

Copy

Users copy a selected run's details by clicking the **Copy** button. Copy the Json Input Parameters to execute a new Routine Method Run by setting the **Input Method** to Json instead of Interactive when configuring your new method run.

Routine Runs

Copy Run Details

Name

sourcedata_20260624172926799

Method Name

source_data_analysis

Description

Run ID

sourcedata-20-nu7djm

Executed By user ID

-1

Run Status

completed

Priority

1

Creation Time

6/24/2026 5:29:27 PM

Last Activity Time

6/24/2026 6:06:17 PM

Artifacts Location

instances_/sda-routine-i-ls95lu/runs_/sourcedata-20-nu7djm/artifacts_

Run Parameters

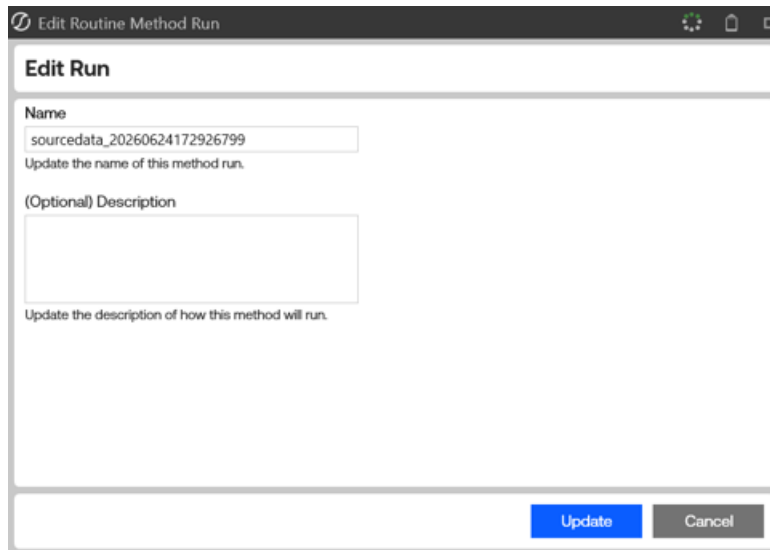
```
{
  "data_connection": {
    "tabular_connection": {
      "connection_key": "sql-server-shared",
      "file_path": "SampleSalesData.csv"
    }
  }
}
```

Close

Edit

Users update the name and description of a Routine Method Run by clicking the **Edit** button.

Routine Runs

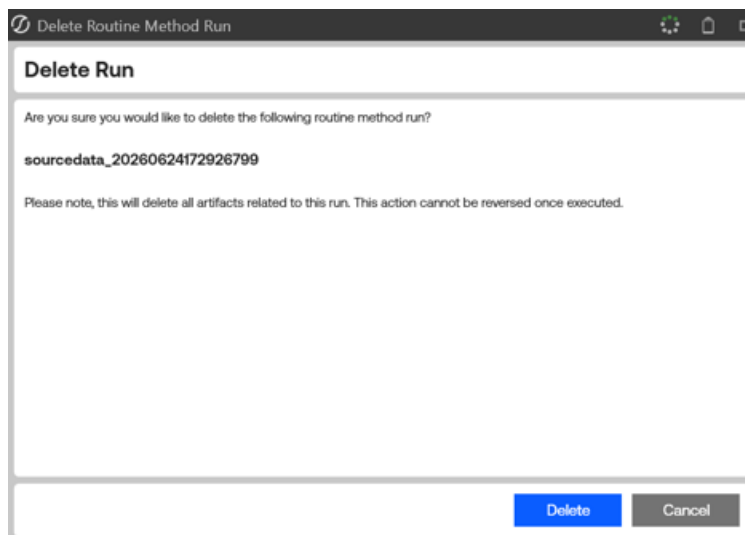


The screenshot shows a dialog box titled "Edit Routine Method Run". Inside, there is a section titled "Edit Run". Under this section, there is a "Name" field with the text "sourcedata_20260624172926799" and a subtext "Update the name of this method run." Below that is an "(Optional) Description" field with a subtext "Update the description of how this method will run." At the bottom right of the dialog box are two buttons: "Update" (in blue) and "Cancel" (in grey).

Delete

Users delete a selected Routine Method Run by clicking the **Delete** button.

CAUTION: All artifacts generated from this Routine Method Run will also be deleted. All deletions cannot be reversed.

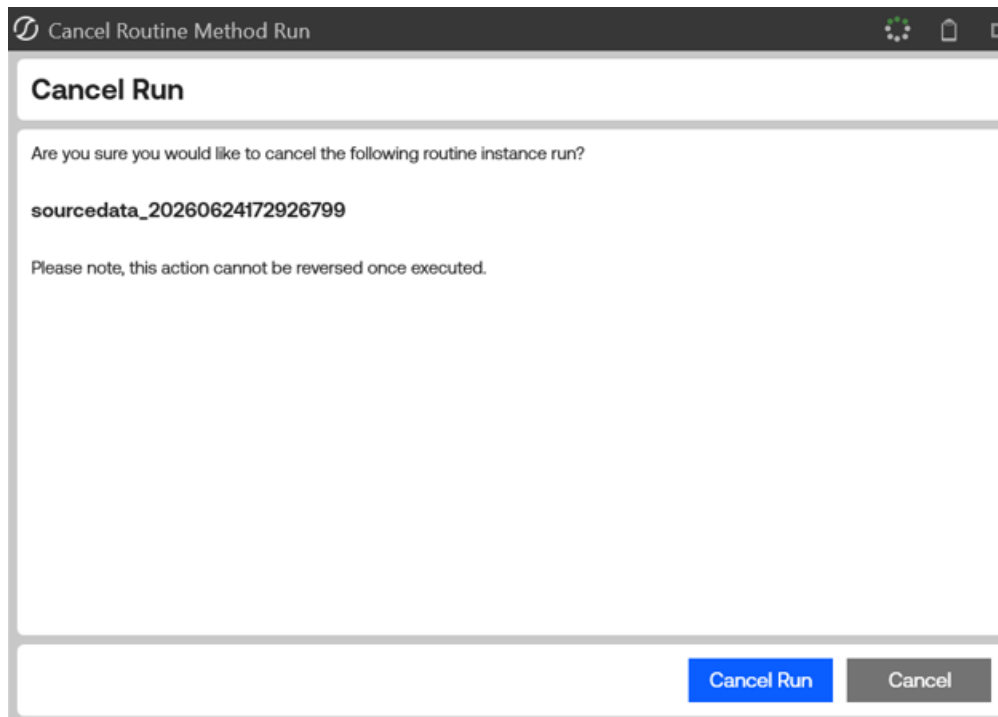


The screenshot shows a dialog box titled "Delete Routine Method Run". Inside, there is a section titled "Delete Run". Under this section, there is a question "Are you sure you would like to delete the following routine method run?" followed by the text "sourcedata_20260624172926799". Below that is a warning message: "Please note, this will delete all artifacts related to this run. This action cannot be reversed once executed." At the bottom right of the dialog box are two buttons: "Delete" (in blue) and "Cancel" (in grey).

Cancel

Users cancel a run once it has started by clicking the **Cancel** button.

NOTE: This action cannot be executed if a run has already completed successfully or entered an error state.



Routine Instance and Routine Run Views

The following views are available for both Routine Instances and Routine Runs: Summary, Artifacts, and Documentation.

Summary

If a Routine Instance has been selected, the Summary view shows additional details about the instance including ID, Creation Time, Created By, and Available Methods.

Routine Instance and Routine Run Views

SDA Routine Instance



Summary

Artifacts

Documentation

Routine	Source Data Analysis
Routine Version	1.0.0
Routine Instance Name	SDA Routine Instance
Description	
Routine Instance ID	sda-routine-i-ls95lu
Creation Time	6/24/2026 5:24:12 PM
Created By	local
Modified Time	6/24/2026 5:24:12 PM
Modified By	local
Memory Override	2
Routine Instance Labels	Data Analysis
Available Methods	Source Data Analysis

If a Routine Run has been selected, users view a summary of the run, including details and a log of run-specific activities. A drop-down menu is provided to access Run Details and the Activity Log.

Routine Instance and Routine Run Views

sourcedata_20260624172926799

100% Completed Task Completed 



Summary

Artifacts

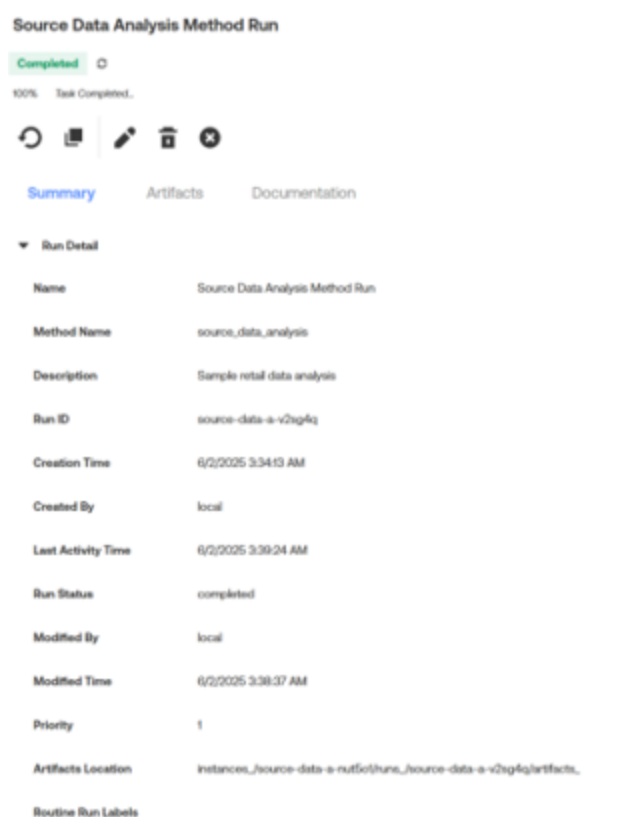
Documentation

▼ Run Detail

Name	sourcedata_20260624172926799
Method Name	source_data_analysis
Description	
Run ID	sourcedata-20-nu7djm
Creation Time	6/24/2026 5:29:27 PM
Created By	local
Last Activity Time	6/24/2026 6:06:17 PM
Run Status	completed
Modified By	local
Modified Time	6/24/2026 6:05:59 PM
Priority	1
Artifacts Location	instances_/sda-routine-i-ls95lu/runs_/sourcedata-20-nu7djm/artifacts_
Routine Run Labels	
Run Parameters	{...}

▼ Activity Log

Captured Time	Log Level	Message
11/13/2024 5:58:41 PM	INFO	100.00% - Final HTML report generated.
11/13/2024 5:58:39 PM	INFO	70.00% - ACF and PACF plots generated.
11/13/2024 5:58:38 PM	INFO	50.00% - Target level plots generated.
11/13/2024 5:55:13 PM	INFO	40.00% - Dataframe pivoted for plot genera
11/13/2024 5:55:13 PM	INFO	30.00% - Summary statistics calculated and
11/13/2024 5:55:13 PM	INFO	1.00% - Dataset loaded from source.



Artifacts

Once a Routine Method Run has successfully completed, view the generated data outputs by clicking on the **Artifacts** tab. An Artifact is a file-based data output that is generated by successfully executing a Routine Method Run. All Methods of a Stateless Routine will produce at least one Artifact; certain Methods of Stateful Routines may produce zero Artifacts. Examples of Artifact output types include parquet files, HTML files, and PDFs.

NOTE: If a user has selected a Routine Instance rather than a Routine Method Run, the Artifacts view will display artifacts from all routine method runs associated with that routine instance.

Routine Instance and Routine Run Views

Using our Time Series Data Analysis example, to view an Artifact:

1. Click on the Routine Method Run from the Routine Run Activity table.
 - a. Wait for it to complete. Monitor the activity status and refresh until the method is completed – may take several minutes.
 - b. If a syserror occurs, refer to the **Activity Log** drop-down menu within the Summary section on the right-hand side of the page. It is possible there is a misconfiguration made in certain column selections when stepping through the component workflow. Repeat steps above as needed.
2. When the routine run completes, click on the **Artifacts** button on the right-hand side of the page.
 - a. Navigate to the artifact named Generic or Advanced Time Series Report.
 - b. Within the files section, click the file called html_content.html.

A box appears containing the actual HTML file.

The screenshot displays the SensibleAI Studio interface. On the left, the 'Library' sidebar is visible. The main area is divided into two panels. The left panel, titled 'Routine Instances', shows a table with columns 'Instance Name', 'Created By', and 'User Name'. Below this is a 'Routine Runs' section with a table showing 'Routine Run Name' and 'Routine Run Method Name'. The right panel, titled 'KalmanFit_20260611154647172', shows the 'Artifacts' tab. It displays a list of available artifacts, including 'Cleansed Data' and 'New vs Old Plot'. The 'Cleansed Data' artifact is selected, showing its details: Name, Qualified Key, Creation Date, and Description. The description states: 'The Kalman cleansed data including the original values and the Kalman filtered values.' The 'Technical Details' section shows the type as 'polars.dataframe.frame.DataFrame'.

Documentation

The documentation section contains the same Routine Documentation as the Explore page, which is dependent on the Routine Instance or the Routine Run that is selected.

The screenshot displays the SensibleAI Studio interface. On the left, the 'Library' tab is active, showing a list of 'Routine Instances' and 'Routine Runs'. The 'Routine Instances' table lists several instances, including 'KalmanFilt_20260611154647172'. The 'Routine Runs' table shows runs for various routines, including 'KalmanFilt_20260611154647172 Constructor'. On the right, the 'KalmanFilt_20260611154647172' routine is selected, and the 'Documentation' tab is active. The documentation page for 'Kalman Filter V2' is displayed, featuring a 'Definition' tab, a 'Signature' tab, and a 'Developer Docs' tab. The 'About' section describes the Kalman Filter as a tool for cleansing time series data, and the 'Tags' section lists 'Data Cleansing', 'Data Preprocessing', 'Time Series', 'Regression', 'Statistics', and 'Optimization'.

Routine Instances

Instance Name	Created By User Name	Descr
KalmanFilt_20260616134021704	local	
KalmanFilt_20260616133955062	local	
PrincipalC_20260615160809755	local	
KalmanFilt_20260611154647172	local	
KmeansClus_20260610165608666	local	
ForecastAI_20260609180548859	local	
NL2Q Testing	local	

Routine Runs

Routine Run Name	Routine Run Method Name
predict_20260611190653144_20260611_150943	predict
predict_20260611190653144	predict
fit_20260611175755332_20260611_141925	fit
fit_20260611175755332	fit
fit_20260611175511464	fit
fit_20260611172854584	fit
KalmanFilt_20260611154647172 Constructor	_init_

Kalman Filter V2 Version: 100

About

A Kalman Filter for cleansing time series data.

The Kalman Filter excels in cleansing time series data by predicting and correcting estimates based on noisy measurements. It iteratively updates predictions as new data arrives, balancing the predicted state against new measurements to filter out noise and refine accuracy. This makes it ideal for applications like financial time series, where it filters erratic data to reveal underlying trends. By modeling both the process and measurement noise, the Kalman Filter efficiently separates the signal from the noise, enhancing data analysis and decision-making accuracy in dynamic environments.

Tags: Data Cleansing, Data Preprocessing, Time Series, Regression, Statistics, Optimization

Package Specifications and Usage Limits

There are two packages available for SensibleAI Studio: Essentials and Standard.

SensibleAI Studio "Essentials" Package

Description

The "Essentials" Package provides AI tools and capabilities to accelerate and enhance your experience around the usage of SensibleAI Forecast and AI Powered Planning processes.

Offering Includes

- SensibleAI Studio Solution
- SensibleAI Studio "Essentials" Routines SensibleAI Studio "Essentials" Solutions
 - Ex: AI Data Tools
 - Ex: AI Time Series Anomaly Detector
- "Essentials" Genesis Blocks
 - Ex: Genesis Block - AI Routine Runner

Usage Limits

Definition	Interaction Locations
A maximum of 3000 Invocations per calendar quarter of LLM Chat Models surface through SensibleAI Studio and Xperiflow Business Rules.	• Xperiflow Business Rules: XperiflowAIClient.PostLLMChatResourceAsync • Xperiflow Business Rules: Xperiflow.LanguageModels.LLMChatClient.InteractAsync
A maximum of 300,000 LLM Embedding Generations per quarter.	• Xperiflow Business Rules: XperiflowAIClient.PostLLMEmbeddingResourceAsync • Xperiflow Business Rules: Xperiflow.LanguageModels.LLMEmbeddingClient.GenerateEmbeddingsAsync • Xperiflow Business Rules: Xperiflow.LanguageModels.LLMEmbeddingClient.GenerateEmbeddingAsync • SensibleAI Studio: “Text Embeddings” Routine

SensibleAI Studio "Standard" Package

Description

The "Standard" Package includes all SensibleAI Studio "Essentials" capabilities as well as additional SensibleAI Studio Powered Solutions, and Genesis Blocks.

Offering Includes

- SensibleAI Studio Solution
- SensibleAI Studio "Standard" and "Essentials" Routines SensibleAI Studio Powered "Standard" and "Essentials" Solutions
 - Ex: SensibleAI Clustering Analysis, AI Data Tools, AI System Diagnostics, AI Account Recs, AI Cube Commentary
- "Standard" and "Essentials" AI Genesis Blocks
 - Ex: AI Commentary Summarizer, AI Routine Runner, AI NL to Table

Usage Limits

Definition	Interaction Locations
A maximum of 3000 Invocations per calendar quarter of LLM Chat Models surface through SensibleAI Studio and Xperiflow Business Rules.	• Xperiflow Business Rules: XperiflowAIClient.PostLLMChatResourceAsync • Xperiflow Business Rules: Xperiflow.LanguageModels.LLMChatClient.InteractAsync

Definition	Interaction Locations
A maximum of 300,000 LLM Embedding Generations per quarter.	• Xperiflow Business Rules: XperiflowAIClient.PostLLMEmbeddingResourceAsync • Xperiflow Business Rules: Xperiflow.LanguageModels.LLMEmbeddingClient.GenerateEmbeddingsAsync • Xperiflow Business Rules: Xperiflow.LanguageModels.LLMEmbeddingClient.GenerateEmbeddingAsync • SensibleAI Studio: “Text Embeddings” Routine
A maximum of 100,000 Code Scan Units per quarter.	• SensibleAI Studio: “Code Scanner” Routine • AI System Diagnostics

Definitions

Routine: A modular software component comprised of one or more functions that accept inputs and produce outputs, such as data results or interactive web applications.

SensibleAI Studio Powered Solution: Any OneStream Solution that has integrated one or more Routines from SensibleAI Studio.

Invocation: A single call or execution of a service, function, or model within the platform.

LLM Embedding Generation: A single operation where an LLM embedding model processes one discrete sub-set of text to produce its numeric embedding representation.

Scan Item: A specific code file or script selected for analysis within a code scanning process.

Condition: A predefined coding pattern or issue (e.g., "Unnecessary Logging Statements", "Nested SQL loops") that the scanning process searches for within a Scan Item.

Code Scan Unit: A single evaluation that checks a Scan Item against one defined Condition, returning any matching instances within that file.